Towards Semantics-Preserving Model Migration

Markus Herrmannsdoerfer and Maximilian Koegel

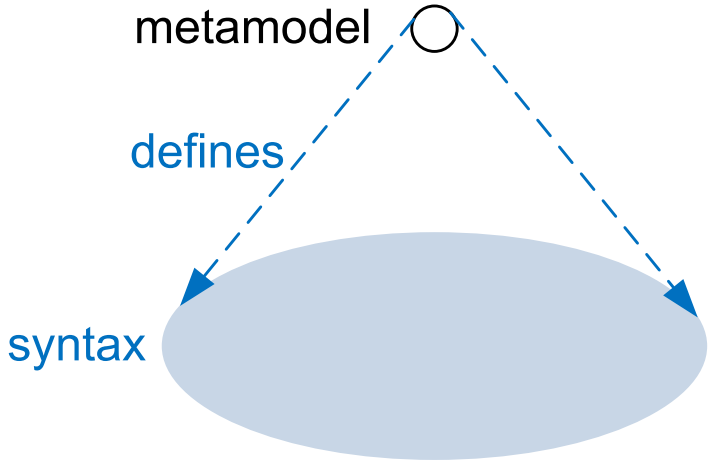
Institut für Informatik
Technische Universität München
{herrmama, koegel}@in.tum.de

Workshop on Models and Evolution
Oslo, Norway
3rd October 2010

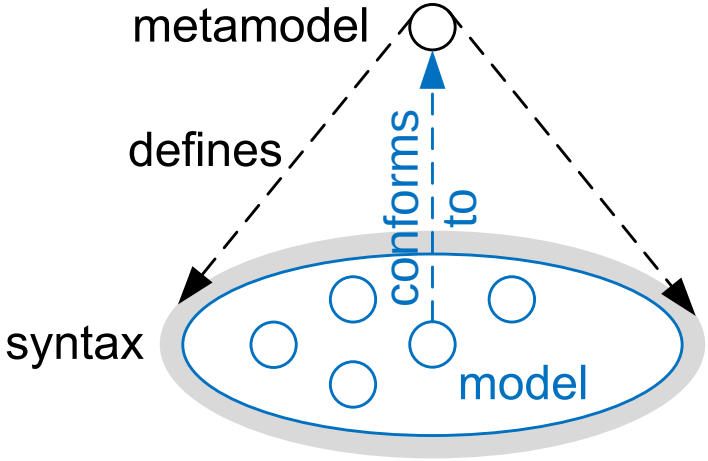
What kind of model(s) do you consider in your work?

metamodel ◯

What kind of model(s) do you consider in your work?



What kind of model(s) do you consider in your work?

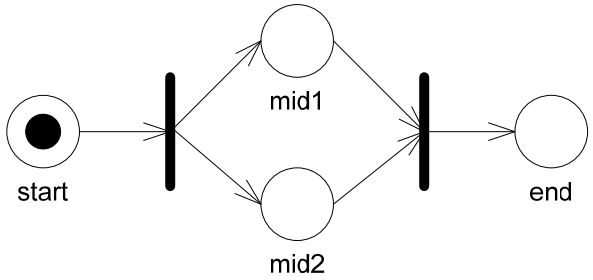


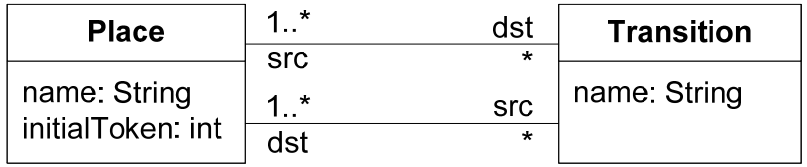
Example: Petri nets – Version 1

Model

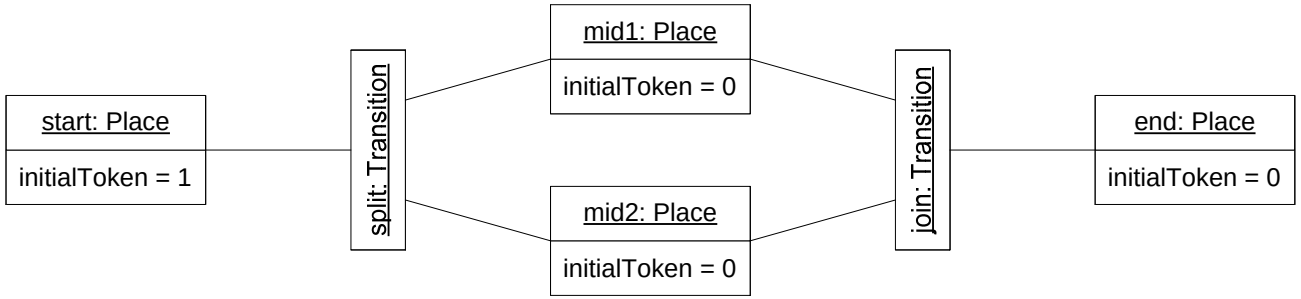
Metamodel

Concrete Syntax





Abstract Syntax

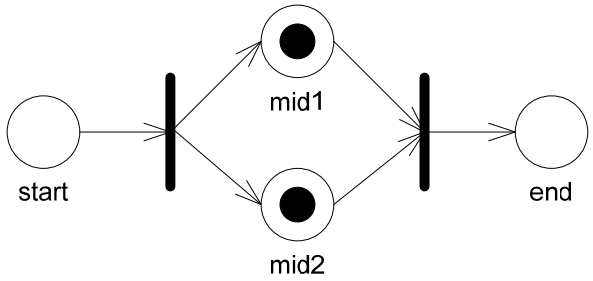


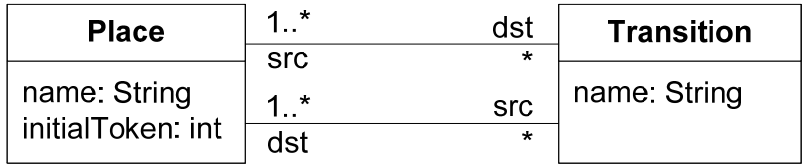
Example: Petri nets – Version 1

Model

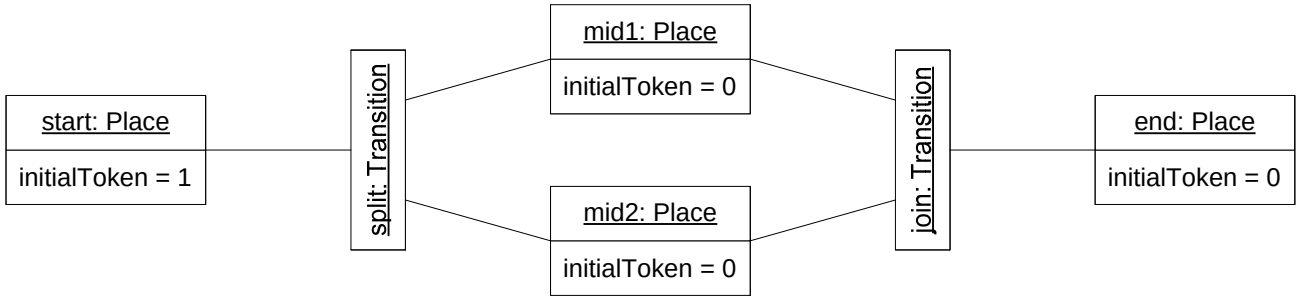
Metamodel

Concrete Syntax





Abstract Syntax

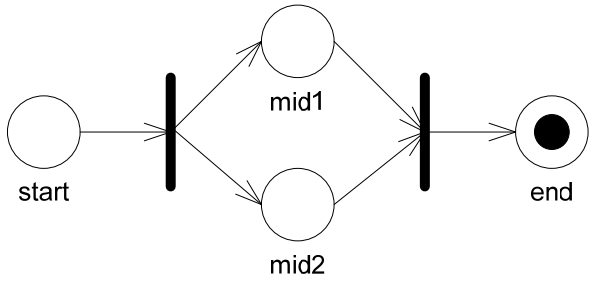


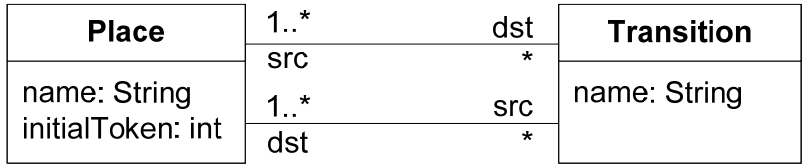
Example: Petri nets – Version 1

Model

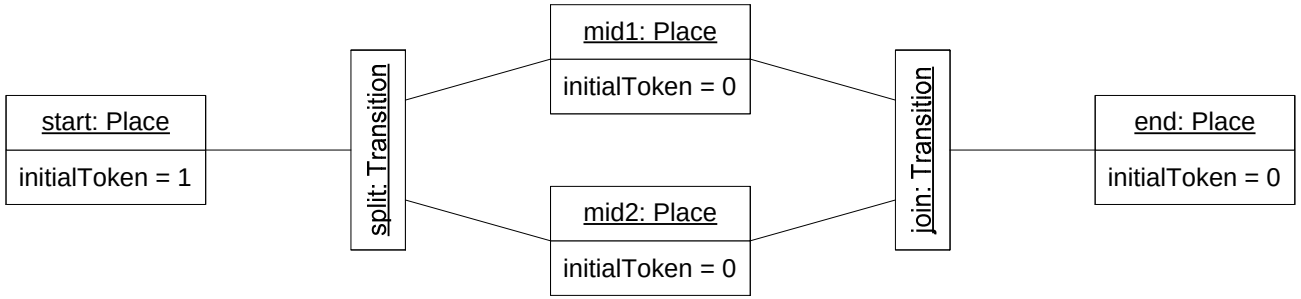
Metamodel

Concrete Syntax

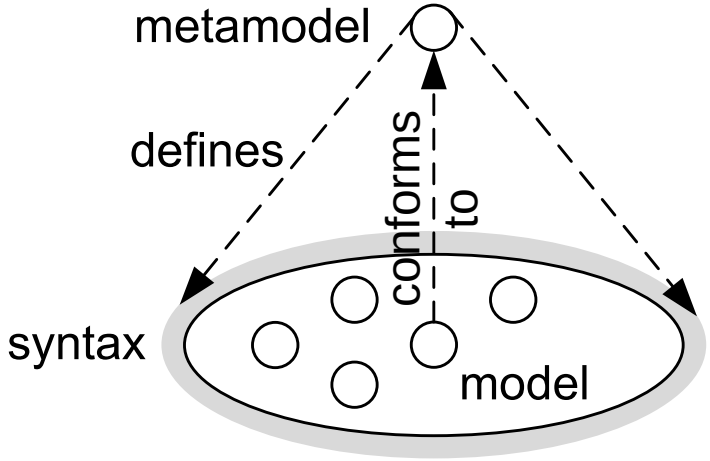




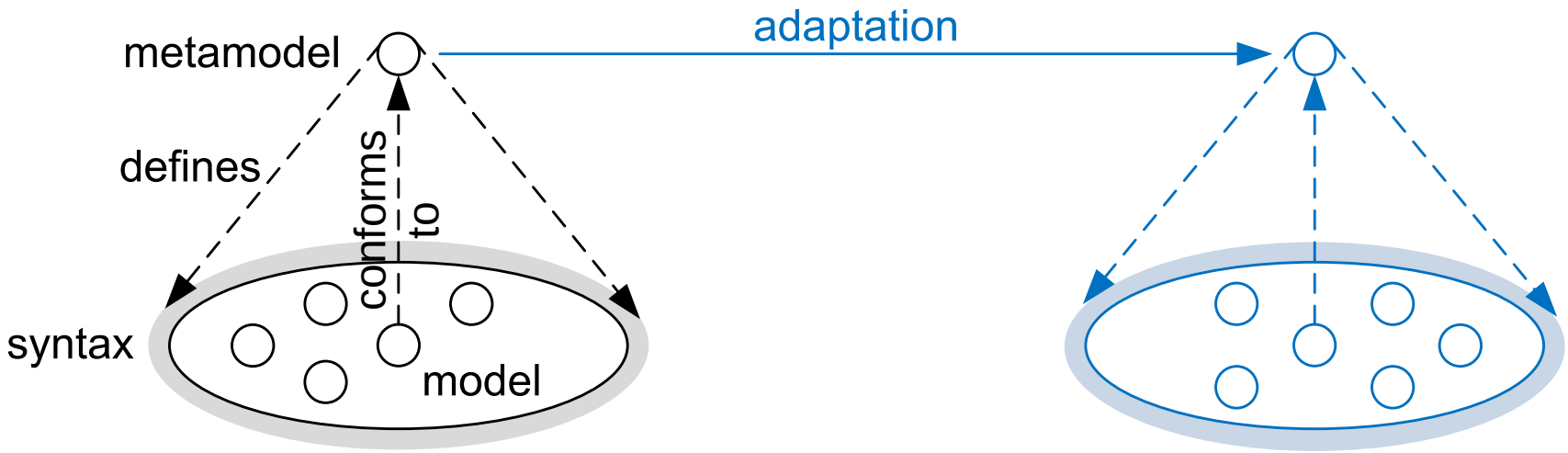
Abstract Syntax



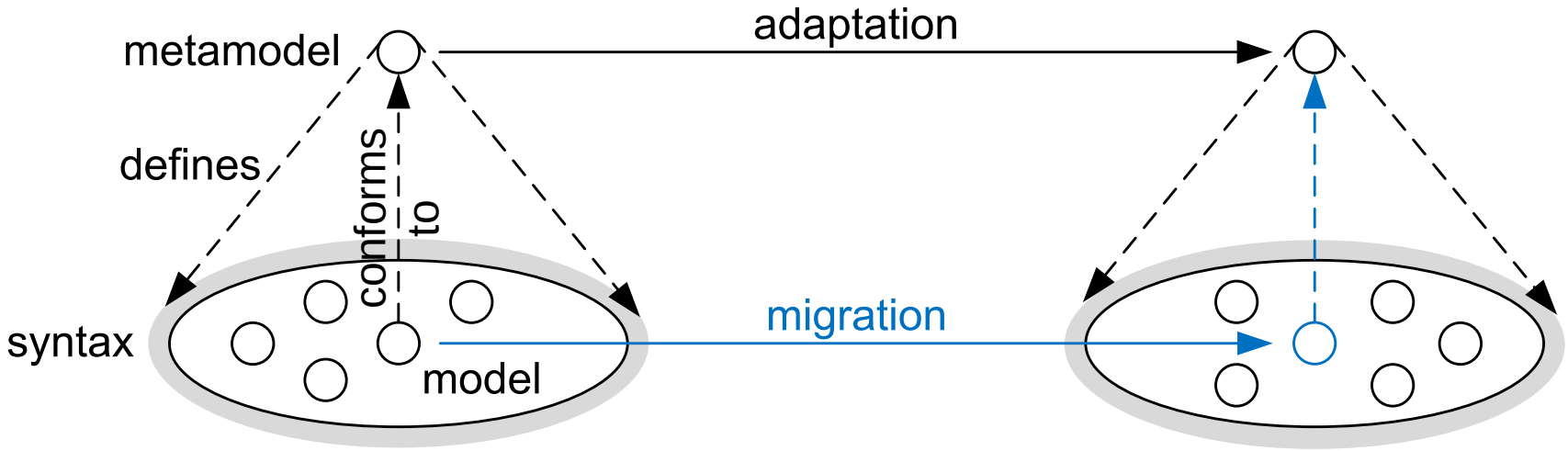
What kind(s) of evolution challenges are addressed?



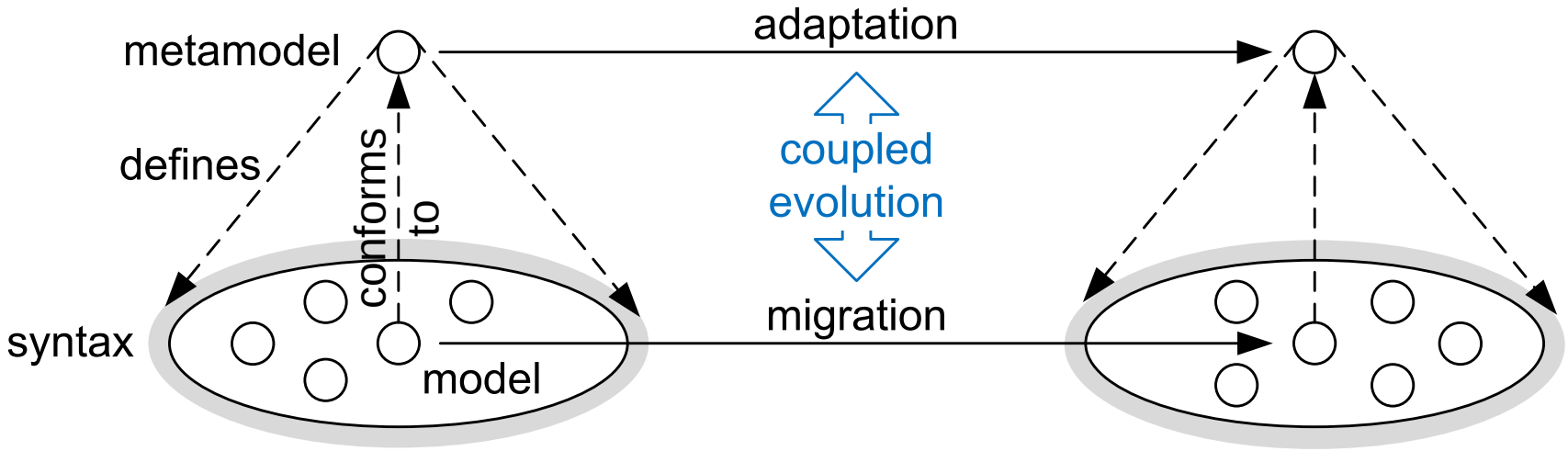
What kind(s) of evolution challenges are addressed?



What kind(s) of evolution challenges are addressed?

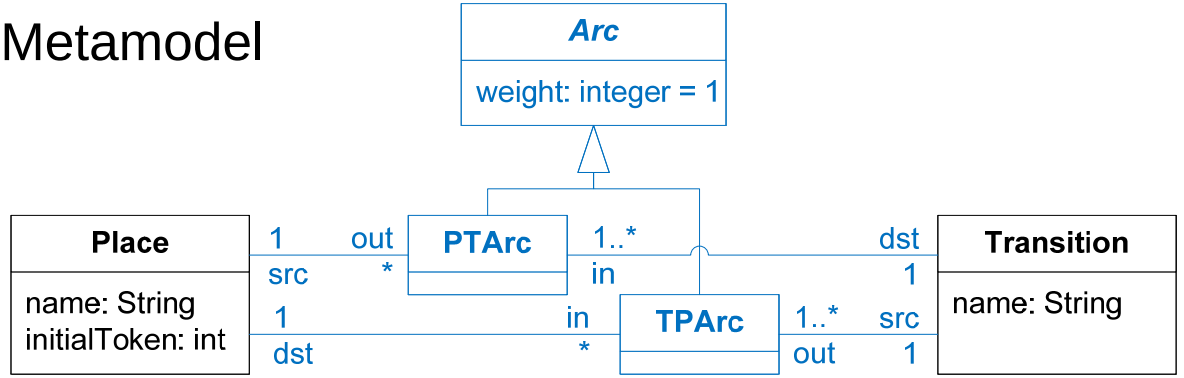


What kind(s) of evolution challenges are addressed?

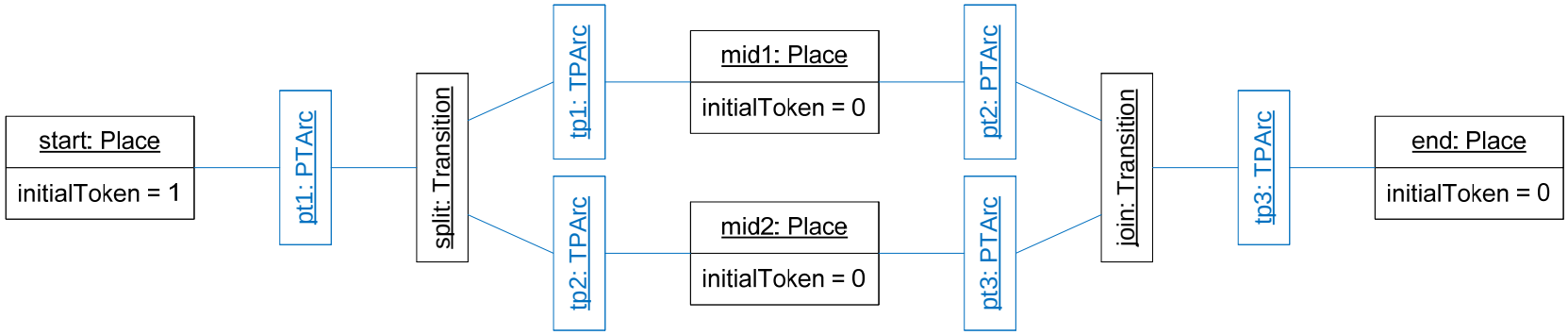


Example: Petri nets – Version 2

Metamodel

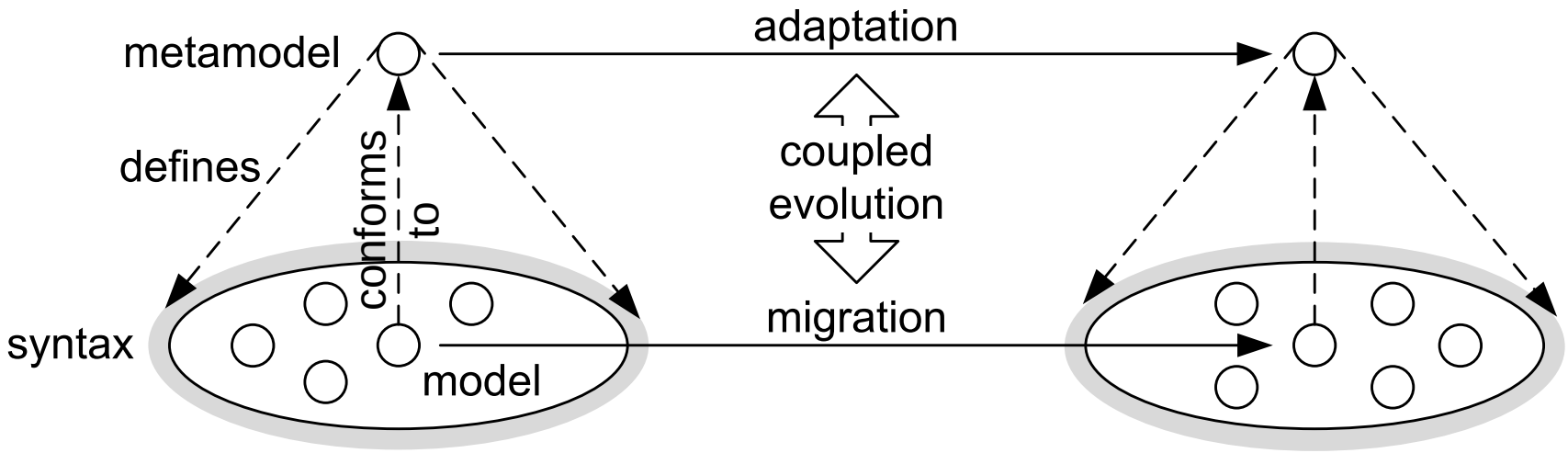


Model



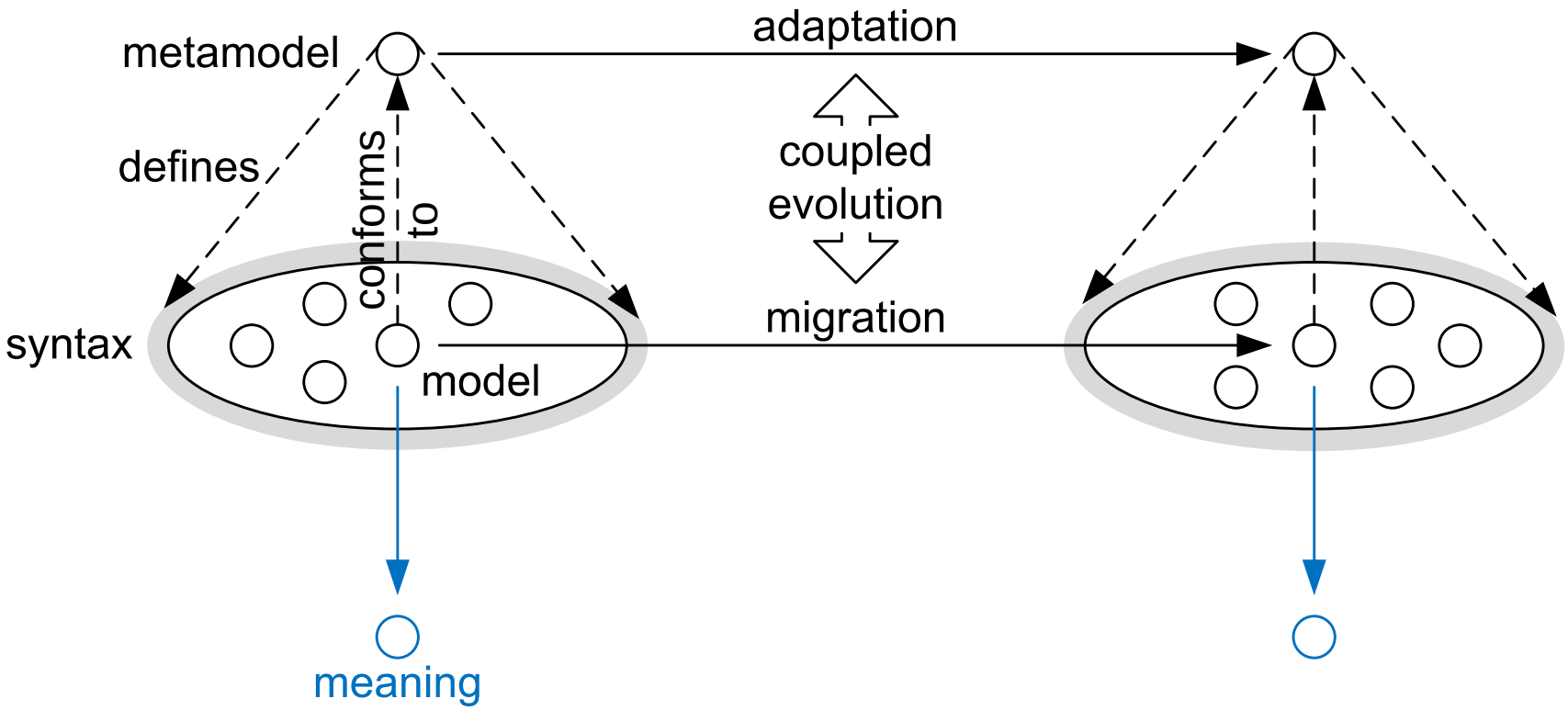
[Wachsmuth. Metamodel adaptation and model co-adaptation. ECOOP 2007]

What kind(s) of evolution challenges are addressed?



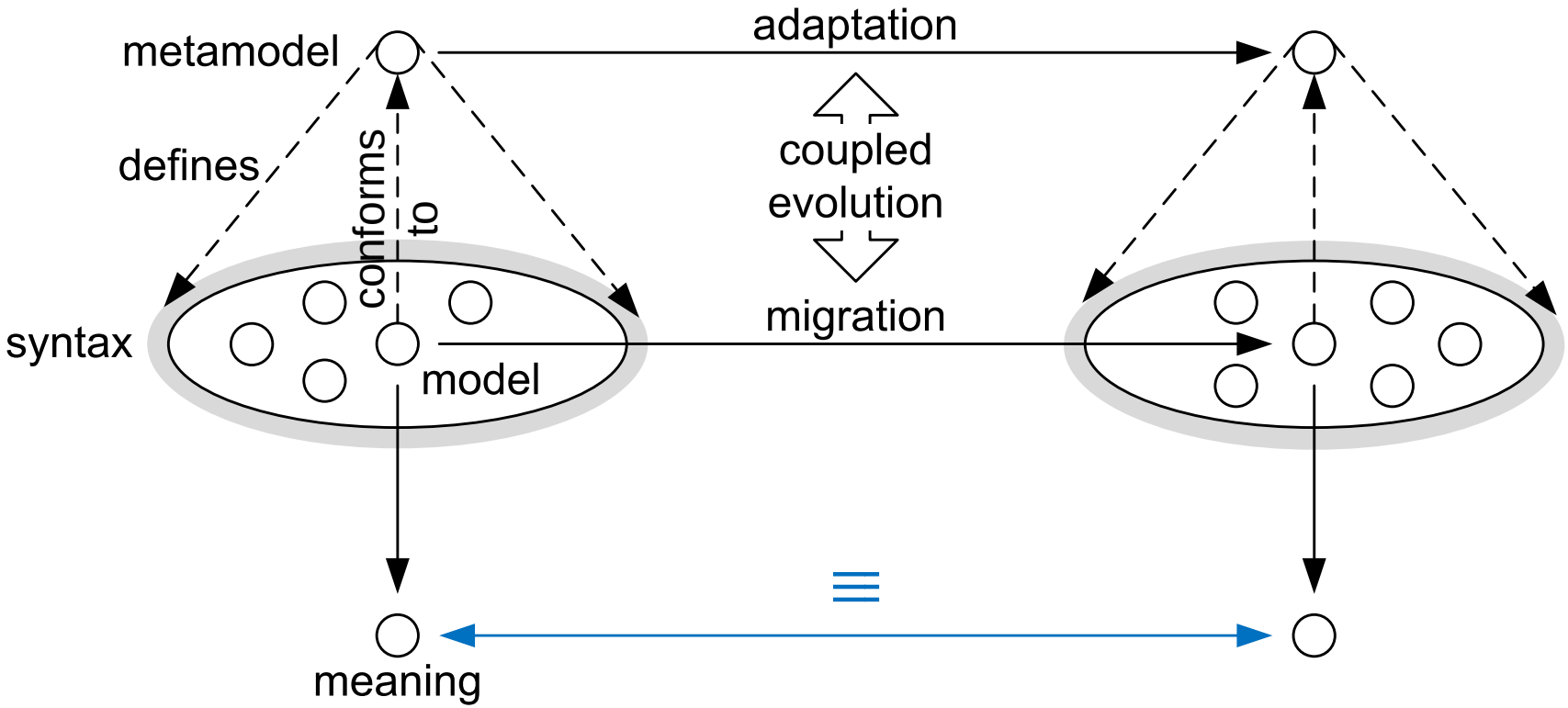
Syntax-Preserving Model Migration

Problem: Semantics-Preserving Model Migration



[Herrmannsdoerfer and Ratiu. Limitations of Automating Model Migration in Response to Metamodel Adaptation. ME 2009]

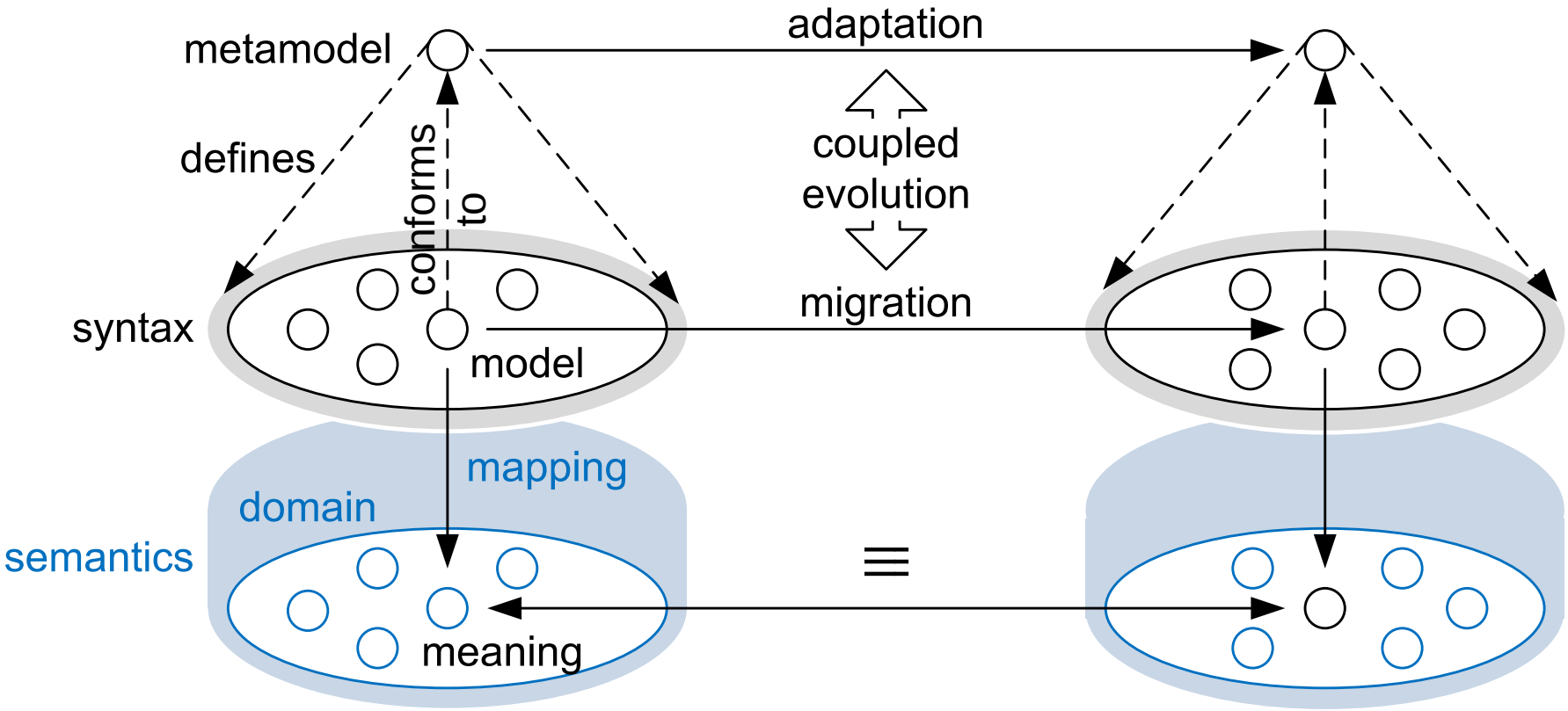
Problem: Semantics-Preserving Model Migration



Otherwise, meaning might be lost during migration

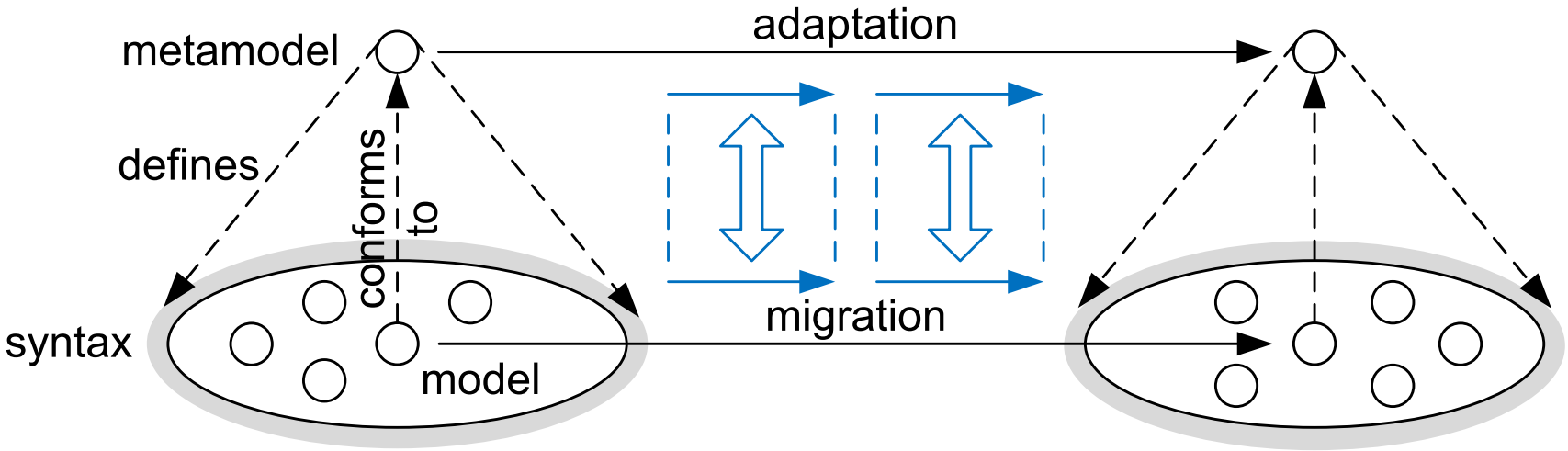
[Herrmannsdoerfer and Ratiu. Limitations of Automating Model Migration in Response to Metamodel Adaptation. ME 2009]

Problem: Semantics-Preserving Model Migration



[Herrmannsdoerfer and Ratiu. Limitations of Automating Model Migration in Response to Metamodel Adaptation. ME 2009]

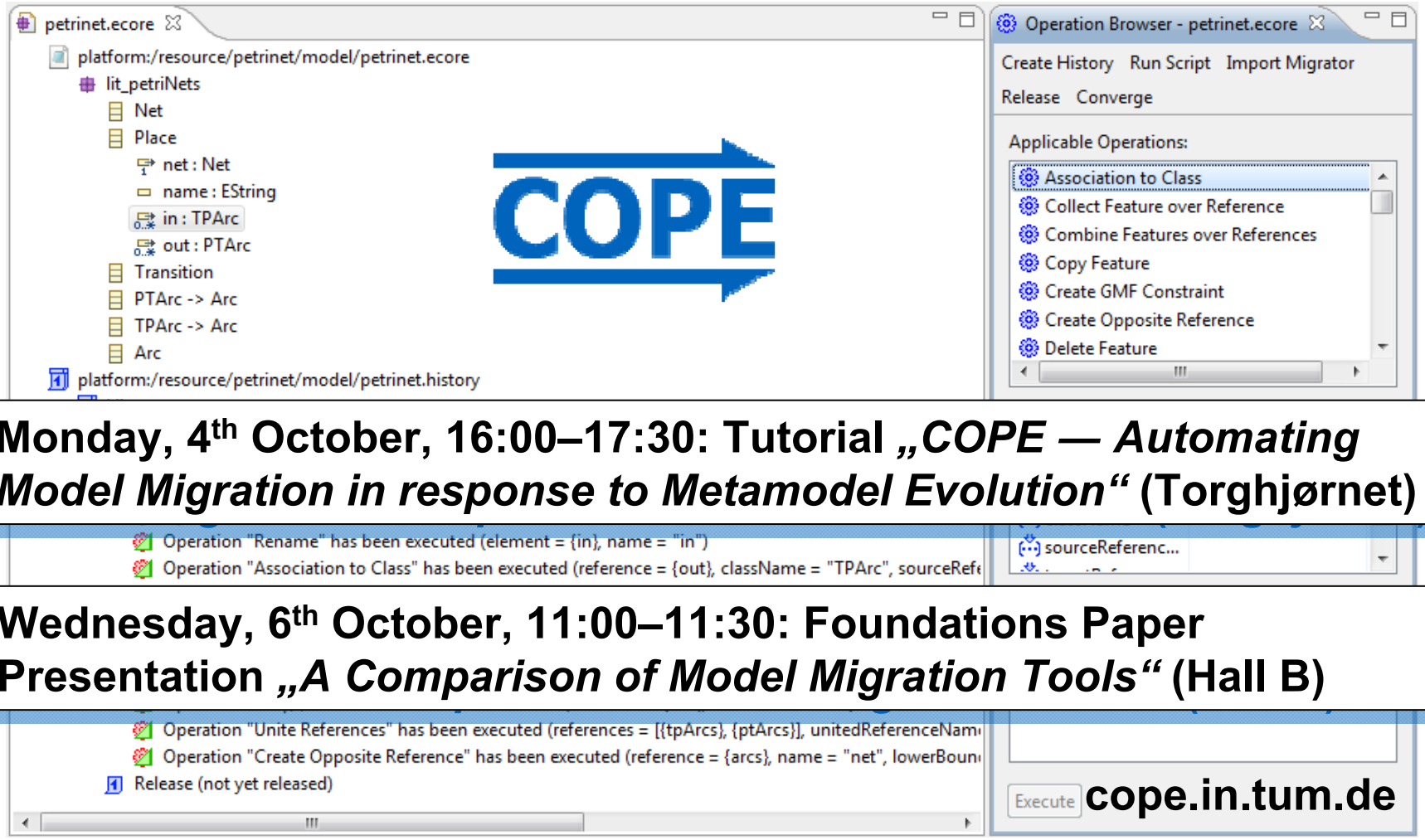
Approach: Operation-based Coupled Evolution



- preserve migration together with adaptation
- specify model migration incrementally
- enable reuse of recurring model migrations

[Herrmannsdoerfer et al. COPE – Automating Coupled Evolution of Metamodels and Models. ECOOP 2009]

Implementation: Eclipse Modeling Framework (EMF)

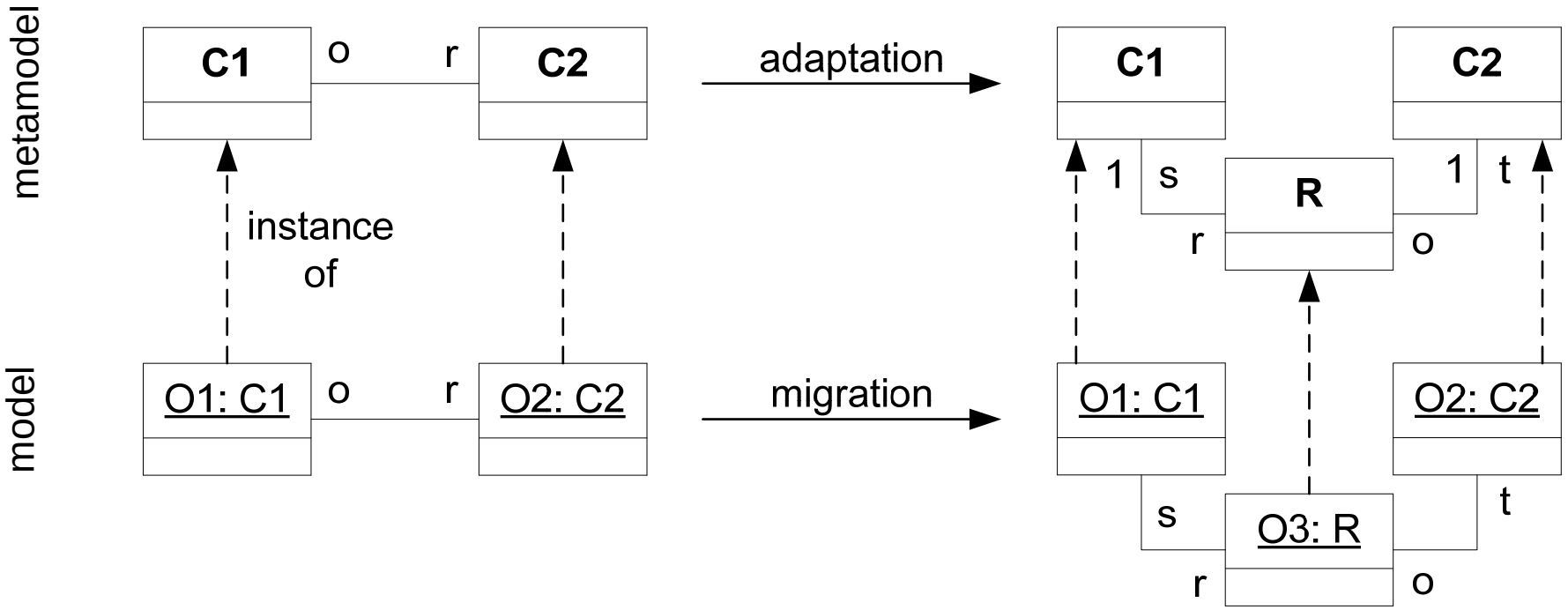


COPE

Monday, 4th October, 16:00–17:30: Tutorial *„COPE — Automating Model Migration in response to Metamodel Evolution“* (Torghjørnet)

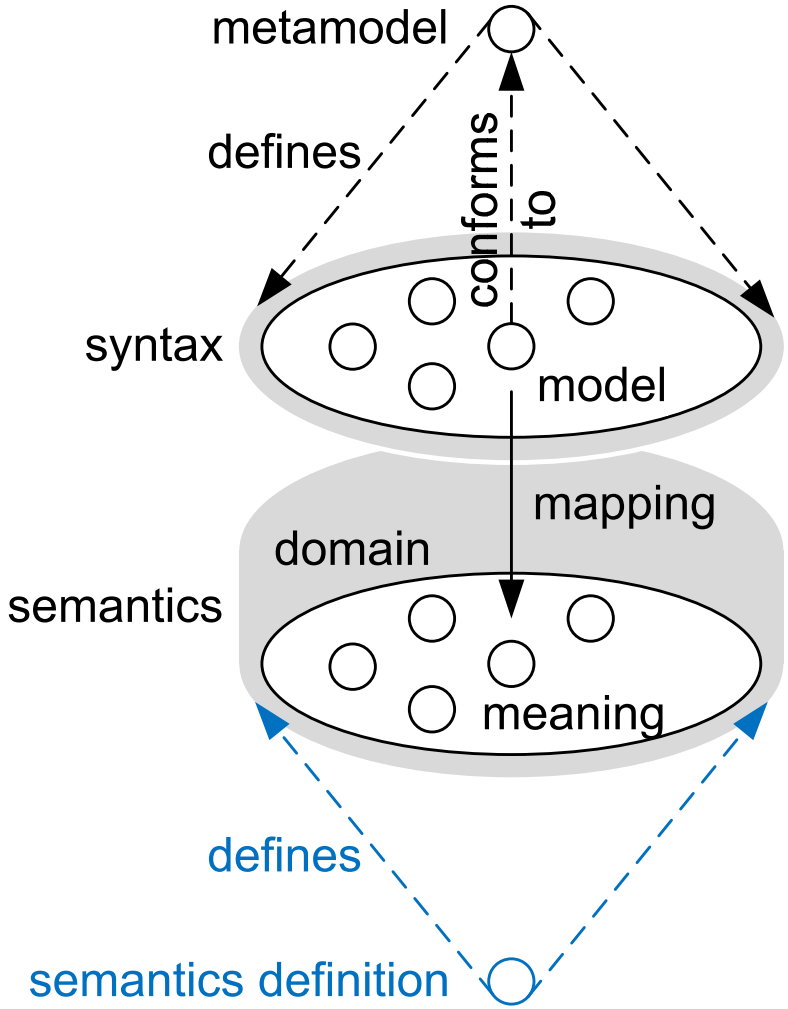
Wednesday, 6th October, 11:00–11:30: Foundations Paper Presentation *„A Comparison of Model Migration Tools“* (Hall B)

Example: Coupled Operation „Association to Class“



Syntax-Preserving
Coupled Operation

Solution: Explicit Model of the Semantics



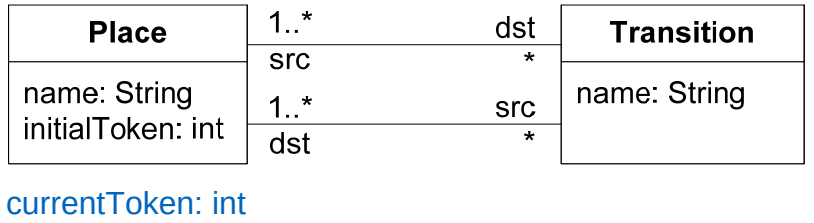
Example: Petri nets – Version 1

Semantics Definition

...

```
Transition.isActivated(): boolean
    return src.every{p ->
     p.isActivated()}

Place.isActivated(): boolean
    return currentToken >= 1

Transition.fire():
    src.each{p -> p.decrement()}
    dst.each{p -> p.increment()}
```

Metamodel

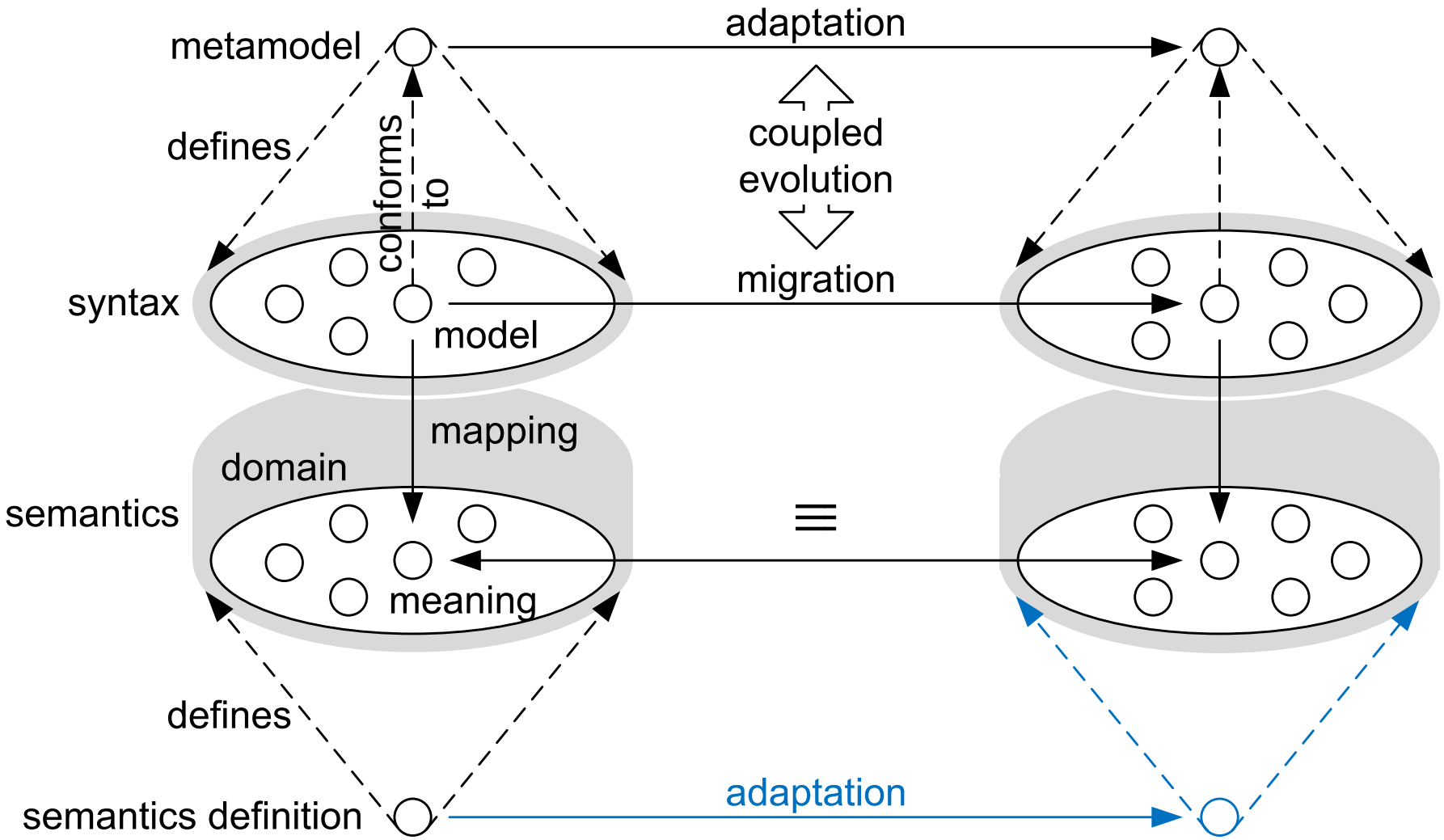


Operational Semantics

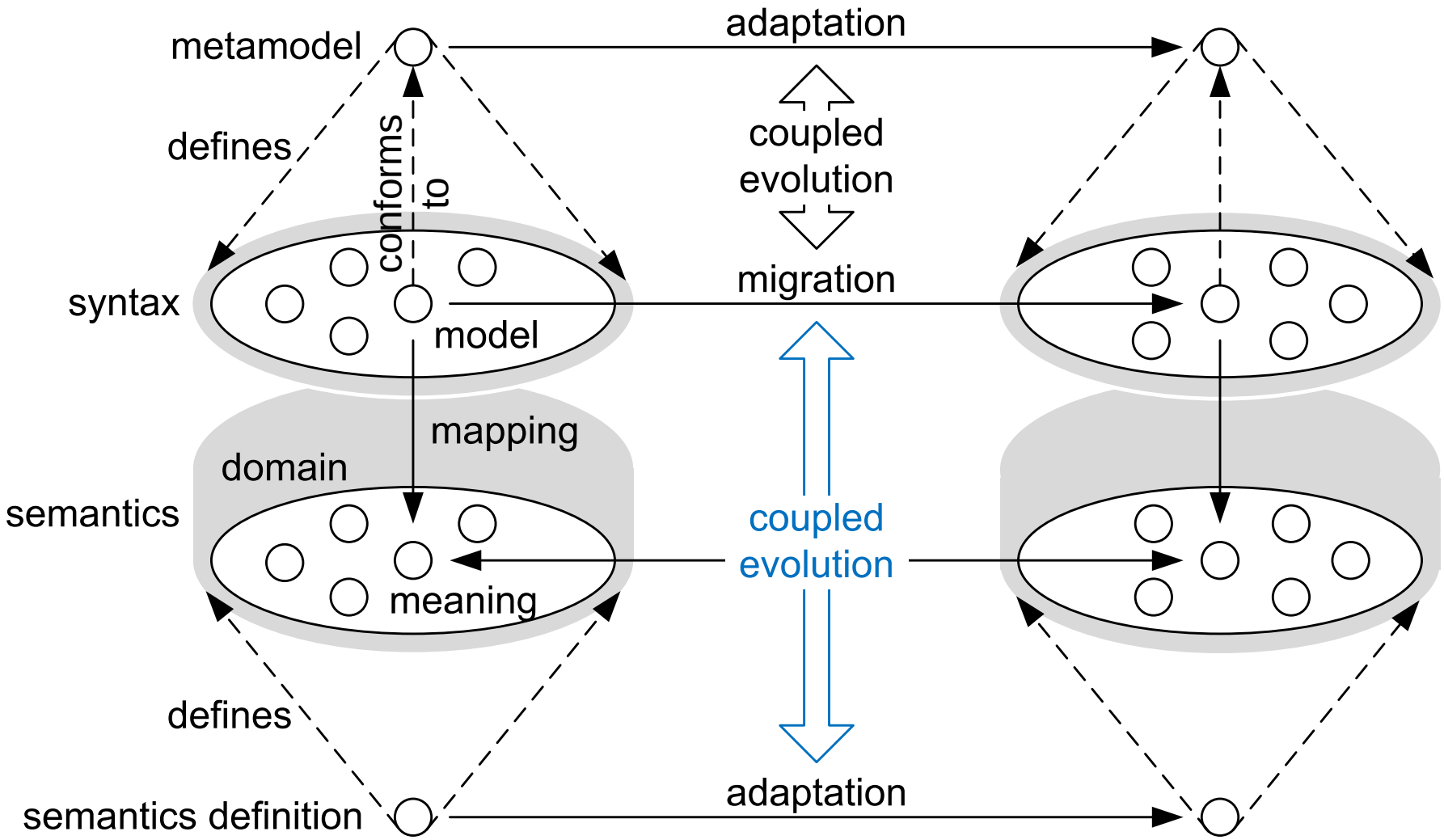
```
Place.decrement():
    currentToken = currentToken - 1

Place.increment():
    currentToken = currentToken + 1
```

[Muller et al. Weaving executability into object-oriented meta-languages. MoDELS 2005]

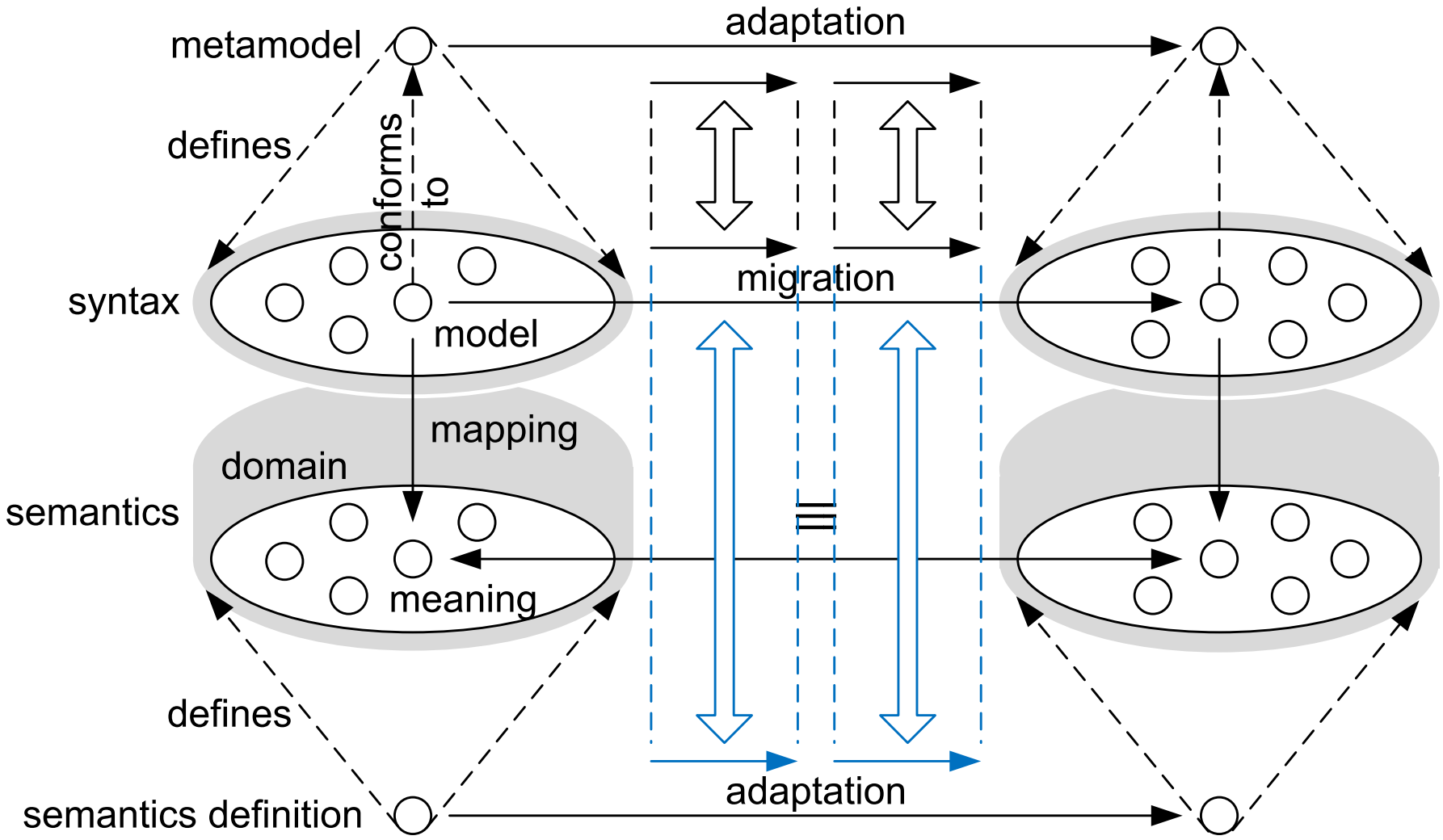
Solution: Adaptation of the Semantics Definition



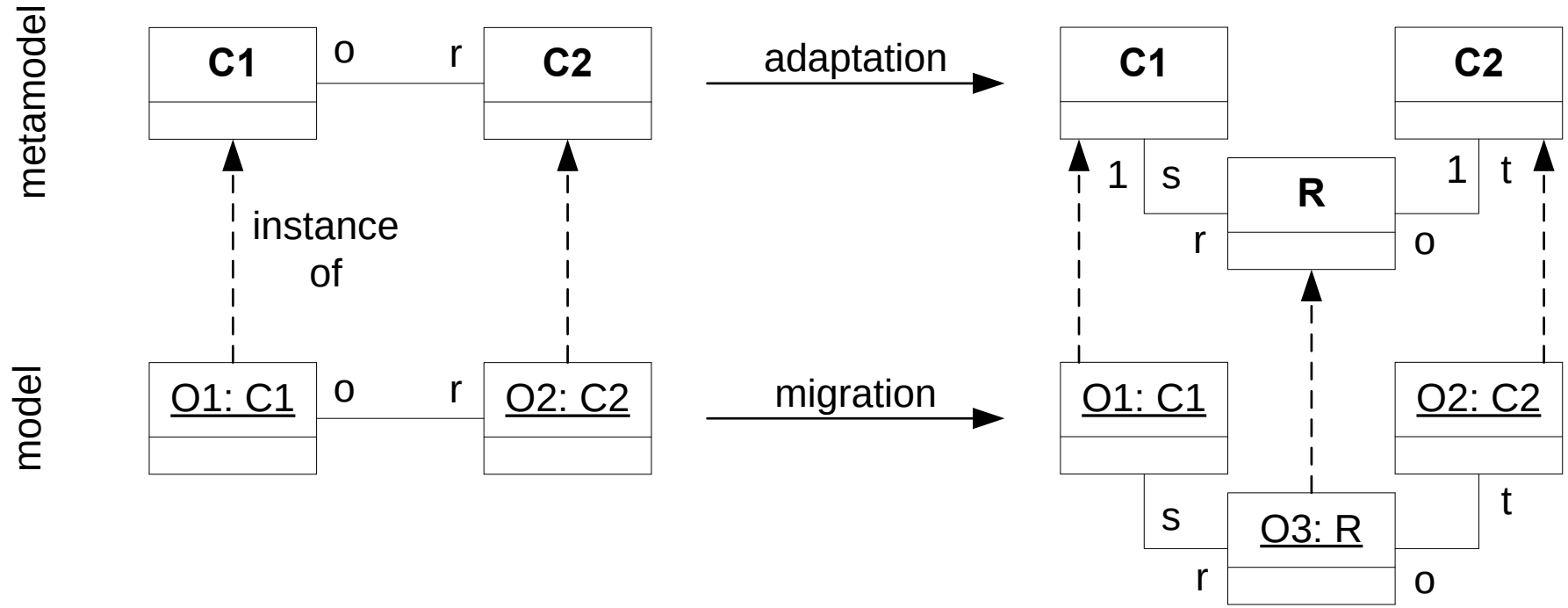
Solution: Adaptation of the Semantics Definition



Solution: Semantics-Preserving Coupled Operation



Example: Coupled Operation „Association to Class“



semantics definition

a.r (a instance of C1)

b.o (b instance of C2)

adaptation

a.r.collect{r -> r.t}

b.o.collect{r -> r.s}

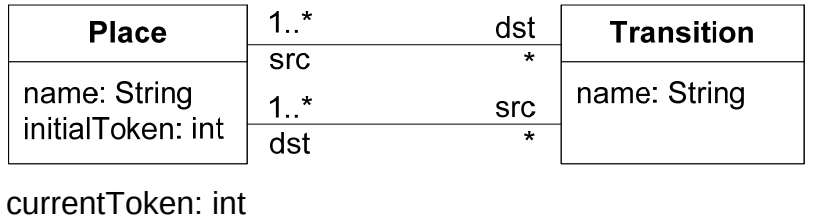
Example: Petri nets – Version 1

Semantics Definition

...

```
Transition.isActivated(): boolean
    return src.every{p ->
     p.isActivated()}

Place.isActivated(): boolean
    return currentToken >= 1

Transition.fire():
    src.each{p -> p.decrement()}
    dst.each{p -> p.increment()}
```

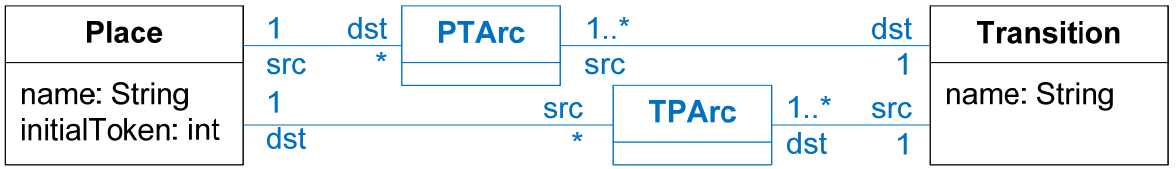
Metamodel



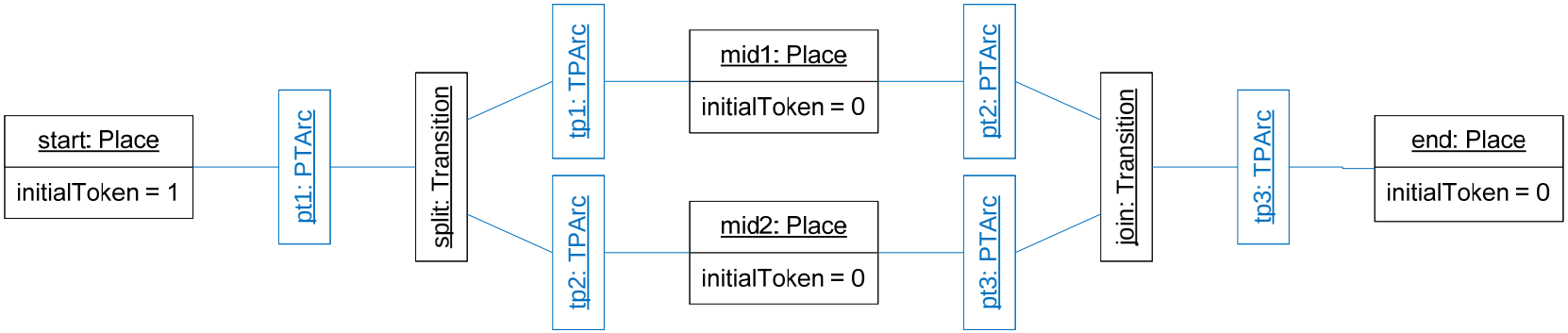

```
Place.decrement():
    currentToken = currentToken - 1

Place.increment():
    currentToken = currentToken + 1
```

Example: Petri nets – „Association to Class“

Metamodel adaptation

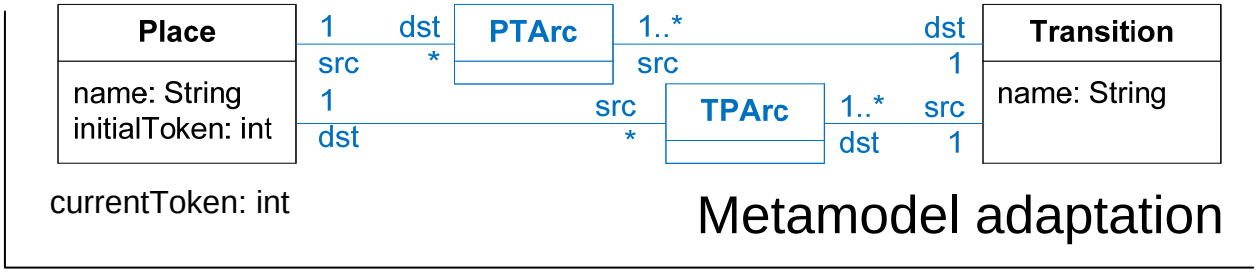


Model migration



Example: Petri nets – „Association to Class“

…



Adaptation of semantics definition

```
Transition.isActivated(): boolean
    return src.collect{pt -> pt.src}.
     every{p -> p.isActivated()}

Place.isActivated(): boolean
    return currentToken >= 1

Transition.fire():
    src.collect{pt -> pt.src}.each{p ->
     p.decrement()}
    dst.collect{tp -> tp.dst}.each{p ->
     p.increment()}
```

```
Place.decrement():
    currentToken = currentToken - 1

Place.increment():
    currentToken = currentToken + 1
```

Case Study: Petri nets Evolution

Coupled Evolution

1. Association to Class (2x)
2. Folding Expressions (Refactoring of the Semantics Definition)
3. Rename (4x)
4. Switch Composite (2x)
5. Extract Superclass
6. Move Operation (Refactoring of the Semantics Definition)
7. New Attribute (using default value)

⇒ **Only Reusable Coupled Operations required**

⇒ **Semantics-Preservation completely automated**

Conclusion: Semantics-preserving Model Migration

Semantics Preservation: consistency between

- model migration
- adaptation of semantics definition

$\Rightarrow$ encapsulate in coupled operation

Coupled Operations: degree of automation

- *reusable*: independent of a specific metamodel
  - semantics preservation can be shown independently of the concrete metamodel
  - covers most of the cases in practice
- *custom*: defined for a specific metamodel
  - semantics preservation has to be shown for the specific case
  - very rare in practice

[Herrmannsdoerfer et al. Automatability of the Coupled Evolution of Metamodels and Models. MoDELS 2008]

Conclusion: Semantics-preserving Model Migration

Future Work: implementation of the approach

- Expressive language for defining the adaptation of the semantics definition
- Define adaptation of the semantics definition for each reusable coupled operation in the library (> 60 operations)
- Analyze the coupled evolution of a real-life modeling language to evaluate the approach
- Provide guidelines to prove semantics preservation for custom coupled operations

[Herrmannsdoerfer et al. An Extensive Catalog of Operators for the Coupled Evolution of Metamodels and Models. SLE 2010]

Open Questions

1. How can we deal with the various metalanguages for semantics definition used in practice?

2. How to ensure semantics preservation in case of a modeling language with implicit / domain semantics (e.g. quality modeling)?

3. Does model migration always need to be semantics-preserving?